The Eighth International Conference on Intelligent Technologies (InTech'07)
Sydney, Australia, 12-14 December 2007

An efficient strategy for robot navigation in cluttered environments in the presence of dynamic obstacles

Jaime Valls Miró, Tarek Taha, Dalong Wang, Gamini Dissanayake and Dikai Liu
ARC Centre of Excellence for Autonomous Systems (CAS)
Faculty of Engineering, University of Technology Sydney (UTS), NSW2007, Australia
Email:{j.vallsmiro,t.taha,d.wang,g.dissanayake,d.liu}@cas.edu.au

***Abstract*— A novel method which combines an optimised global path planner with real-time sensor-based collision avoidance capabilities in order to avoid moving obstacles (e.g. people) in a complex environment is presented. The strategy is based on a time efficient one step path planning algorithm for navigating a large robotic platform in indoor environments. The planner, which has been proved to compare favourably to currently available path planning algorithms such as Randomly-exploring Random Trees (RRTs) and Probabilistic Road Maps (PRMs) in known static conditions, is enhanced here with a modified Variable Speed Force Field (VSF^2) mechanism to accommodate for dynamic changes of the environment. The basic concept of the modified $DVSF^2$ is to generate a continually changing parameterised familiy of virtual force fields for the robot based on characteristics such as location, travelling speed, heading and dimension of all the objects present in the vicinity, static and dynamic. The interactions among the repulsive forces associated with the various obstacles provide a natural way for local collision avoidance and situational awareness. This is harnessed here by locally modifying the planned behaviour of the moving platform in real time, whilst preserving as much as possible the optimised nature of the global path. Furthermore, traversability of the path is continually monitored by the global planner to trigger a complete re-planning from the robot's current location in the case of major changes to the environment, most notably when the path is completely blocked by an obstacle. Overall, a complete solution to the navigational problem in partially known cluttered environments is provided.**

I. MOTIVATION

The problem of planning and tracking a path in a partially known cluttered environment in the presence of other moving objects is considered here. Classically, the simpler problem of planning in static environments has been approached from two perspectives, opposite to some extent, yet both reliant on the use of advanced perception abilities to gather as much information as possible about the environment:

- Use the sensorial data to build or modify a detailed internal representation of this environment which the robot would then use to (re)plan a feasible path from source to the destination.
- Employ the measurements in a reactive obstacle avoidance controller.

The former is the most generalised technique to address the problem in highly structured environments as it follows the intuitive sense-model-plan-act technique which more closely resembles the human reasoning, and a vast number of solutions can be found in the literature (see [4] for some examples). However its performance in real scenarios is poor given the unreliable nature of the perceived data and the high computational cost of modelling complex environments. Moreover any changes in the robot surroundings require a reiteration of the algorithm or the pre-computed path will almost certainly hit obstacles, hence making it particularly unsuitable for dynamic environments.

These perceived limitations have led to mobile robot navigation modules where locally simple reactive behaviours are in charge of computing the motion of the robot. Rather than storing a detailed model of the world to compute a geometric path free of collisions with the perceived obstacles, the technique effectively uses the world as its own model, introducing sensory information within the control loop. The main drawback then becomes its locality in execution [37], as not all information is available to the robot's sensors. For this reason, while some form of reactive obstacle avoidance behaviour becomes necessary (particularly in dynamic environment), the use of a high-level world model and a global planning strategy is also mandatory. This hybrid model has emerged as the preferred motion planning architecture for mobile platforms in use today.

In our previous work [2] we addressed the problem of designing an efficient near optimal global planner for mobile platforms. The proposed solution was proved to be time efficient and feasible even in the case of large robots in narrow passages, and compared favourably to currently available static path planning algorithms such as RRTs and PRMs. However, the uncertainties the system was capable of dealing with were limited to those a simple linear controller could overcome in bringing the robot back to the pre-planned path. The absence of an interactive response to changes in the environment meant the solution was limited to known static environments. In the work presented here these shortcomings are overcame by introducing a novel reactive planner in the loop in line with the hybrid approach scenario described.

A. Contribution

The main contribution of this paper lies with the framework proposed to ensure safe and optimal navigation of a robot in uncertain dynamically changing scenarios. Given a partially known model of the environment, a higher-level planner [2] is employed for describing an optimised path from source to destination, and also for constantly monitoring the traversibility of the waypoints to trigger a global re-planning when the feasible path gets obstructed. An innovative strategy is then presented to handle underlying local navigation issues in between waypoints. The methodology, framed within the context of a force field approach recently proposed in the literature for multi-robot path planning and collaboration [3], is suitably modified here to handle the behaviour of the robot at a local scale in the presence not only of static changes and uncertain localisation within the environment, but also of moving objects in the vicinity. In the approach, a virtual force field is constructed around the robot based on the current characteristics from the moving platform itself such as travelling speed, dimension and location, combined with the status of the surrounding dynamic and static obstacles as given by the sensorial information. It is the interaction of all the attractive and repulsive forces at play what drives the robot towards its local goal in an efficient and natural manner. This is demonstrated with the simulation of an autonomous wheelchair operating in a challenging office-like scenario with dynamic obstacles moving nearby.

The remainder of this manuscript is organised as follows: latest proposals to the path planing and obstacle avoidance problem and where our approach represents an improvement is analysed in Section II. The methodology used for the creation of the search space and the actual path finding is summarised in Section III, with Section IV focusing on the proposed force field reactive planner. Detailed simulation results are given in Section V. Finally, Section VI summarises the contributions of this paper and future work.

II. PATH PLANNING AND OBSTACLE AVOIDANCE BACKGROUND

A. Path Planning

Motion planning is characterised by the ability to compute a collision-free path from an initial pose to a goal position in between static obstacles, or by maintaining a set of constraints in the state of the world such as following a target or exploring unknown environments. Motion planning has been extensively studied and the reader is referred to [4] for a thorough review. Most of the current approaches are based on the concept of configuration space (C-space) introduced by Lozano-Pérez [10] and Wesley [11] in which robots are regarded as a single point in a high dimensional space equivalent to the degrees of freedom of the robot. As the obstacles are generally expanded in an over-simplistic manner by the length of the larger robot dimension, this approach very often prevents larger robots in cluttered environment from ever reaching a solution, even when it exists [10].

The exact construction of the C-space is a computationally expensive solution [12] and sampling-based techniques, in which a random set of points are used to represent the C-space, have been develop which provide a faster practical solution by sacrificing completeness [15], [18]. Traditionally, sampling-based algorithms are based on uniform sampling which considers the whole environment as uniformly complex and thus the overall sampling density will be equivalent to that needed by the most complex region, reaching its worst case scenario when narrow passage areas exist in the environment [13]. Non-uniform methods such as the Gaussian sampling strategy [16] and the bridge test [14] have been proposed to ensure that most of the configuration space is actually close to obstacles or inside a narrow passage, thus reducing unnecessary samples and decreasing the computational time.

Sampling algorithms are generally divided into two based on the complexity of the pre-processing step heuristics which set up the search space: multi-query and single-query approaches. The former starts with a step that usually takes a large amount of time but makes solving path planning problems in the same environment faster. Probabilistic Road Maps (PRMs) [15] is an example of a multi-query approach that initially used uniform sampling in constructing the path, although nowadays PRMs are moving into non-uniform sampling methods [24]. Single-query methods were developed to avoid the large pre-computational time that the multi-query methods take, and they have been proved to be efficient [17]. Randomly-exploring Random Trees (RRTs) [18], [19], mainly based on single-query methods, have gained popularity from their good performances. This has lead to a number of extensions specifically targeted to the solution of complicated geometrical problems [20], such as the deterministic resolution-complete alternatives that have been proposed to replace the random sampling methods in [21].

In many cases, an optimal and not just a feasible path is required. The random nature of the above planners means the paths generated are very often sub-optimal and non-smooth, not particularly suitable for realistic robot traversal. A two-phase approach was proposed in [22] to optimise paths in the special case where the first-phase generated paths are made up of straight line segments connected by way-points. Another two-phase planning algorithm based on RRT was developed in [23] where low cost paths are computed by a numerical gradient descent algorithm that minimises the Hamiltonian of the entire path. In this work, a hybrid time-efficient path planning algorithm developed by the authors which has proved to compare favourably to PRMs and RRTs [1] has been employed. The strategy, based on a non-uniform multi-query planner, uses an a-priory static map of the environment to calculate an offline, minimalistic free search space where computational resources are directed towards the narrow passages. The algorithm takes further advantage of techniques like the bridge test and an optimised obstacle expansion method to further reduce the number of samples and the points to be check for obstacle collision. The optimality and smoothness weaknesses of probabilistic path planners is addressed by the addition of a smooth cost function to the search space. A modified A* search is then implemented to find suitable paths on this space. The result is a time-efficient path planner with smooth and cost optimal paths.

B. Obstacle Avoidance

Local obstacle avoidance methods focus on changing the robot's trajectory based on its sensors data during robot motion. The "bug" algorithm [27], which proposes to circumnavigate the obstacles by following the contour of each obstacle in the robot's way is perhaps the simplest obstacle avoidance algorithm, where no control, kinematic or dynamic constraints are considered. Techniques such as potential field methods [28] are particularly attractive too as their simple conception can be used for on-line purposes to drive the robot among static obstacles towards the target. Here the robot is treated as a point under the influence of an artificial potential field. The basic idea behind all potential field approaches is that the robot is attracted toward the goal, while being repulsed by the sensed obstacles. Although the approach does not take into account robot kinematics, other strategies have been proposed to transform the three-dimensional obstacle avoidance problem with shape and kinematics constraints into the simpler planning scenario of a point moving in a two-dimensional space without kinematic restrictions [37].

However, traditionally the problem is solved in the control space by computing a set of collision-free admissible motions, followed by a selection based on some form of optimisation and convergence towards the target. This is the case of the curvature velocity method (CVM) [34], which takes the actual kinematics constrains and some dynamic constrains into account. The drawback of this method resides in the circular simplifications attributed to the shape of the obstacles, and the existence of local minima. Other techniques which also take into account admissible controls and robot shape constraints include the dynamic window obstacle avoidance method (DWA) [35] and the global dynamic window approach (GDWA) [36] variation.

The family of vector field histogram (VFH) [29] techniques also take into consideration some form of kinematic/dynamic constraints and is a preferred choice in mobile robot applications for its speed and results. The algorithm looks for gaps between the obstacles in front of the robot and builds a local map based on the concept of certainty grid [30] from recent sensor range readings. A variation of the original, the VFH+ [31], first comes up with a simplified model of the moving robot's possible trajectories based on its kinematics limitations. Then obstacles which block the robot's allowable trajectories are properly taken into account in a masked polar histogram. VFH* introduced the A* search into the direction determination and has been proved to obtain better solutions than VFH+ in certain cases [32].

A method which is particularly relevant to the work presented here is the elastic band method proposed in [33]. This technique tries to combine the global path planning with real-time sensor-based collision avoidance. A pre-planned global path is deformed in real time to keep a robot away from obstacles during its movement, while the internal contraction forces will bring the robot back to its original path when the obstacle is out of the sensor range. The method takes also into account the actual shape of the robot and restricts the search space by the concept of a "bubble". A bubble is defined as the maximum local subset of the free space around a given configuration of the robot which can be safely travelled in any direction without collisions. Given such bubbles, a band or string of bubbles can be used along the trajectory from the robot's initial position to its goal position to show the robots expected free space along the pre-planned path.

The obstacle avoidance techniques just mentioned and others variations which follow similar concepts have been proved to be effective in the presence of static obstacles in the environment. However, one aspect they all have in common is that they are not directly applicable to deal with dynamic obstacles, since characteristics from the moving objects such as travelling speed, size, etc. are not built into the algorithms. That is not to say some of them will not be able to deal with a certain degree of dynamism in the environment, but it is not designed in their methodology to do it in an efficient manner. The $DVSF^2$ obstacle avoidance technique proposed here is designed around these premises and performs well in partially known and continuously changing environments. Its basic principle is to generate a repelling force field for the robot based on its own kinematic and dynamic status and those of the moving objects, as well as the the static obstacles to generate a combined force that drives the robot towards the target.

III. THE OPTIMISED GLOBAL PLANNER

In this section an overview of the proposed path planner algorithm is given. Further details can be obtained from [2].

A. Generating the Search Space

The pre-processing step aims at minimising the on-line computation by pre-generating a search space to contain all the information that will be used during the on-line path planning. The steps used during the search space creation can be defined as follows:

Algorithm 1 Search-space generation

Input: map, robot dimensions
1. Expand Obstacles.
2. Generate Regular Grid with low resolution.
3. Apply bridge test to add dense narrow passages.
4. Penalise nodes by adding a smoothing cost.
5. Connect nodes to form search space discretisation.
6. Eliminate those that cause collision.

Output: free search space.

1) Collision Detection: In this step obstacles are enlarged with a radius R to simplify the on-line collision detection by reducing the number of points on the robots to be checked for collision. This is done by finding the largest possible expansion radius R that allows the robot to pass through the narrowest path and then divide the area of the robot into circles of that radius, as depicted in Fig. 1. The centre points of those circles

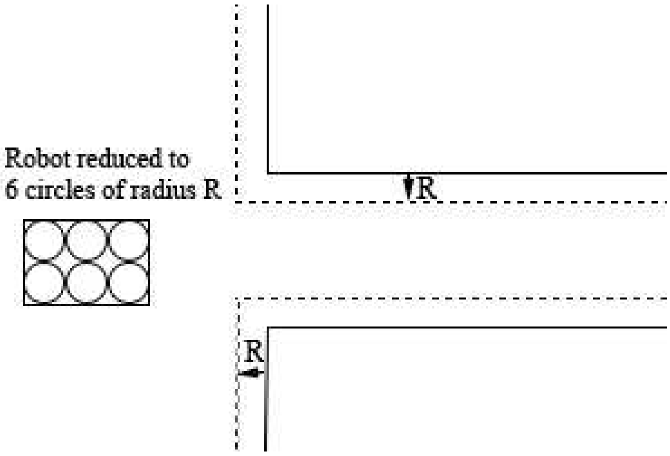


Fig. 1. The area of the robot is covered by circles of radius R, the centres of these circles will be the points to be checked for collision.

will then be used to check for obstacle collision. The expansion radius R is determined based on the a-priory knowledge of the environment: suppose that the narrowest passages is of width l and the largest robot dimension is r then the largest expansion that allows the robot to pass through can be determined by:

$$R = \begin{cases} \frac{l-\varepsilon}{2} & \text{if } l < 2r \\ r & \text{otherwise} \end{cases} \quad (1)$$

where ε is a minimal safety distance to make sure the platform does not get uncomfortably close to the obstacles.

2) Regular Grid Discretisation: The C-space is then populated with nodes using a low resolution regular grid. This will help in maintaining the connectivity of the graph by defining a minimum discretisation for the open spaces. The discretisation density is adjusted to suit the environment, selecting as sparse a grid as possible. Up to this stage the nodes hold only position information.

3) Bridge Test: The bridge test [14] was introduced to boost the sampling density inside narrow passages using only a simple test of the local geometry. A short line segment of length d can sample randomly through a point m in the free space such that the end points of the line segment lie in obstacles. This line segment is what we call a bridge because it acts like a bridge across the narrow passage with its endpoints in an occupied location and the point m in a free space. If we are able to build a bridge through point m, then the bridge test is successful at this point and point m is added to the search space.

4) Clearance (Smoothness) Penalty: To insure that the paths generated are directed to the middle of the empty space not adjacent to the obstacles, we introduced a cost added to the the nodes in the search space indicating how far they are from an obstacle. The cost C_p is a normalized cost that is inversely proportional to this distance d, so that the closer the point is from an obstacle the higher its cost, according to:

$$C_p = \frac{K_d - d}{K_d} \quad (2)$$

Algorithm 2 Bridge test

1. repeat
2. Pick a point p from the regular-grid map
3. **If** p is in an occupied location **then**
4. Pick a point p' that is d distance away from p
5. **If** p' is in an occupied location **then**
6. Let m be the midpoint of pp' line segment
7. **If** m is in a free location **then**
8. Insert m into the search space as a new node

where K_d indicates the clearance distance beyond which the node will be assigned a zero cost. This cost will be used during the on-line path planning process to plan smoother paths in one step.

5) Node Connections: In order to find a path among the resulting nodes, these need to be connected together in a tree structure. This is done by establishing a link between each node and its neighbouring nodes a certain distance away. The more neighbours a node is linked to, the more discrete poses (position and orientation) with be available during the search for a viable path.

6) Connections Reduction: In a final step we eliminate those connections that cause a collision of the platform with the obstacles. The connections between nodes determines the possible orientations of the robot should it follow that path. The centre of the circles that describe the area of the robot along that path can be rotated and translated accordingly. Hence we can then determine if any of them falls into an occupied area or not.

B. On-line Path Planning

In the algorithm proposed here, we use the well know A* algorithm where our cost function $J(d)$ combines the sum of the partial path distances Δd, the sum of all the distances travelled as a result of changing orientation $\Delta\theta\, x$ where x is the length of the axis of the rear wheels, the sum of the clearance penalties C_p previously computed offline - which is directly proportional to the distance Δd, the number of reversals (backward motion) in the path n_{rev} and the heuristic function h representing the distance to goal at each step. The cost function, defined as follows:

$$J(d) = \sum \Delta d + \sum \Delta\theta\, x + \sum \Delta d\, C_p + \sum n_{rev} + h \quad (3)$$

encourages the robot to avoid whenever possible turns and reversing actions, while at the same time directing it towards the middle of free space. The result is a smooth and secure path - in the context of the obstacles around the platform - efficiently generated in a single step.

IV. THE VIRTUAL FORCE FIELD FOR LOCAL PLANNING

This section describes the modified VSF^2 method for local re-planning in the presence of static and dynamic obstacles,

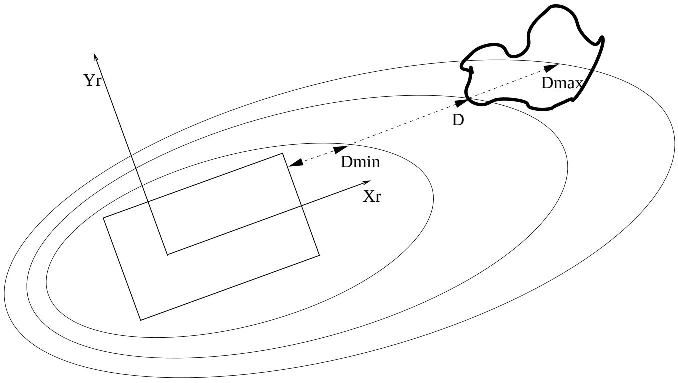


Fig. 2. The repulsive virtual force field generated by an obstacle on the robot.

$DVSF^2$. For more details on the VSF^2 for multi-robot motion planning the reader may refer to [3].

A. *Definition of the Force Field*

A force field is defined as a virtual repulsive force in the vicinity of a robot, whose magnitude and orientation varies constantly with the robot's status as it travels along the workspace towards the goal. Intuitively, it is a force that increases as the sensed distance to any obstacles in the environment decreases, hence keeping the robot from hitting obstacles.

To introduce the concept of the virtual force field is important to understand how the interacting forces from obstacles and robot are calculated. Assuming a rectangular-shape differential-drive platform like the automated wheelchair robot considered here, restricted to move on a flat surface, the contours of the force fields at various distances D are geometrically represented by conic sections of different eccentricities, as depicted in Fig. 2. The robot coordinate system, at its centre of rotation, coincides with the conic section focus, and the robot's X_r axis lies along the conic section's major axis. It is shown how D_{max} describes the maximum distance where the field is felt, whilst D_{min} is the distance at which this robot has maximum repulsive force. Or, in other words, D_{max} represents how far the robot can influence its vicinity, and D_{min} preserves a safe area around the robot onto which no other objects should ever be present. These relationships can be mathematically described in the robot reference system depicted in Fig. 3. Let Rr be the radius, from the robot's origin, of the maximum circle embedding the robot, θ_r the robot orientation in the global coordinates, θ the orientation of the obstacle relative to the robot, and v_r the linear component of the robot speed (v_{max} being its maximum value). Then:

$$E_r = \frac{1}{K_v}\,\frac{v_r}{v_{max}},\ K_v > 1 \tag{4}$$

$$D_{max} = (K_r\,R_r)\,\frac{E_r}{1-(E_r\,cos\theta)},\ K_r > 1 \tag{5}$$

$$D_{min} = K_{d_{min}}\,D_{max} \tag{6}$$

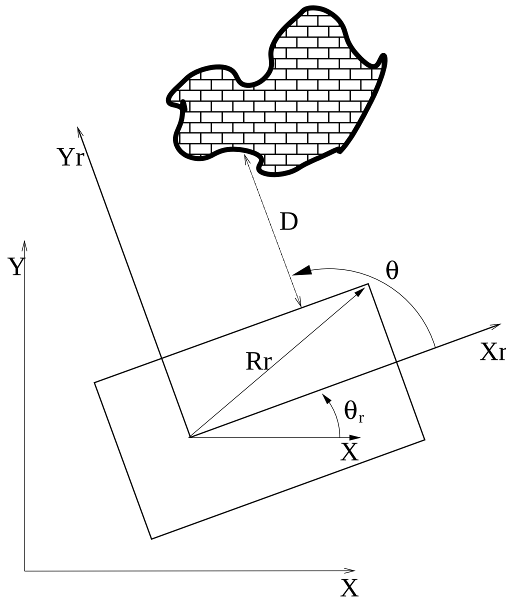


Fig. 3. Reference parameters that define the robot's virtual force field.

K_v is the eccentricity factor required to limit the speed ratio E_r so that when $v_r = v_{max}$ the resulting conic is an ellipse. This is required to guarantee the effect of the field is felt in a limited, closed area around the robot. A parabolic or hyperbolic open ended field would not allow for the proper detection of the obstacles around the robot as it moves about. The factor R_r allows us to take the size of the robot in the calculations of the field, so that bigger platforms generate larger fields, whilst the positive multiplier K_r represents an extra degree of control of the influence that the environment has in the resulting field by permitting larger virtual fields when the area is a-priory known to be fairly clear of obstacles. D_{min} is a percentage of D_{max}, i.e, $0 < K_{d_{min}} < 1$. The choice of $K_{d_{min}}$ denotes to some extent the environment's influence on the force, as it heavily influences how close the robot can get to obstacles. For a large robot such as a wheelchair operating through narrow doors this needs to be fairly small, or the robot will get stuck.

1) Repulsive Force: With the above definitions in mind, the repulsive force generated by a robot at a distance D from its boundary is defined by:

$$|\vec{F}_{rep}(D)| = \begin{cases} 0 & D > D_{max} \\ K_f\,\frac{D_{max}-D}{D_{max}-D_{min}} & D_{min} \le D < D_{max} \\ \infty & \text{otherwise} \end{cases} \tag{7}$$

where K_f is a positive constant which determines the magnitude of the repulsive force. As distance D changes from D_{min} to D_{max} the magnitude of the repulsive force changes gradually from K_f to 0. Equation (7) can be easily transformed to emphasise the concept of the contour of the force field as seen in Fig. 2 by defining

$$\rho = \frac{D}{D_{max}} \tag{8}$$

so that (7) becomes:

$$|\vec{F}_{rep}(\rho)| = \begin{cases} 0 & \rho > 1 \\ K_f \, \frac{1-\rho}{1-\rho_0} & \rho_0 \leq \rho < 1 \\ \infty & \text{otherwise} \end{cases} \tag{9}$$

Now when ρ changes from ρ_0 to 1, the magnitude of repulsive force changes from K_f to 0. With this mapping Eq. (6) becomes:

$$D_{min} = \rho_0 \, D_{max} \tag{10}$$

Given the manifold of repulsive force fields that accompany the robot in its motion, when an obstacle is sensed in the vicinity, the interaction point where the maximum repulsive force is felt is selected as the repelling contour force for the obstacle, $\vec{F}_{rep\_obstacle}(\rho)$.

2) Attractive Force: At any given time a goal is selected as the furthest visible point provided by the sensor along the optimised pre-planned global path. A virtual attractive force $\vec{F}_{attr}$ is associated with this local target to attract the robot from its current location towards the goal. The attractive force is presumed exerted from the centre of rotation of the robot and is calculated as a positive value set to be larger than the sum of all repulsive forces from the obstacles, so that the robot is guaranteed to always being pulled in the broad direction of the target.

B. The Combined Forces

The resulting forces exhorted on the robot can then be calculated by:

$$\vec{F}_{total} = \vec{F}_{attr} + \vec{F}_{rep\_total} \tag{11}$$

where the reaction repulsive force is the sum of the fields from each of the n obstacles - static and dynamic - observed in the local are, as given by

$$\vec{F}_{rep\_total} = \sum_{i=1}^{n} \vec{F}_{rep\_obstacle_i(\rho)} \tag{12}$$

C. The $DVSF^2$ Controller

The motion of a wheelchair-like differential robot in a two-dimensional surface is constrained by the well-known non-holonomic relation:

$$-\dot{x} \, \sin\theta + \dot{y} \cos\theta = 0 \tag{13}$$

whose effect is to reduce the dimension of the location and orientation configuration space $q = (x, y, \theta)$ to a kinematic model whose instantaneous velocity lateral to the heading has to be zero. This is normally expressed in terms of the translational movement v_r, and the rotation movement about its centre of mass ω_r, which are also the controls of the wheelchair robot. Hence, given the interacting forces expressed by Eq. (11), the following motion controls are proposed

$$v_r = \frac{v_{max}}{1 + \frac{max(|\vec{F}_{rep\_obstacle_i}|)}{K_f}} \tag{14}$$

$$\omega_r = f(\vec{F}_{total}, \theta_r) \tag{15}$$

where $f(\vec{F}_{total}, \omega_r)$ is a normalising function of the angular error difference between the desired direction along the resultant force $\angle\vec{F}_{total}$ and the robot's current orientation, while Eq. (14) guarantees that the robot's speed will be sensibly influenced by the obstacles' repulsive forces.

V. EXPERIMENTAL RESULTS

The results of simulating the proposed algorithm in an automated wheelchair platform are presented next. The mechanical platform measures $1.2m \times 0.7m$, by all accounts a large robot when made to manoeuvre in a challenging office-like environment with narrow passages, doors, long corridors and cluttered static obstacles. The wheelchair has two differentially driven wheels at the rear and two passive casters at the front and can travel at speeds of up to 15km/h. The planner was commanded to find an optimal path in a given map, rather noisy as it has been obtained via ICP scan-matching of data collected by driving the real platform around the area. Given a start and goal configurations the planner finds the shortest collision-free path shown in Fig. 4(a). Note the smoothness and how the planner tends to direct the robot as far as possible from obstacles and along the middle of the empty spaces, making the path not only feasible and kinematically "gentle" on the platform, but also maximising clearances with the obstacles known at this stage along the minimal distance path.

Fig. 4(b) depicts the scenario when a static obstacle (red square box) is placed in the robot surroundings as it is navigating along the given path. It can be seen how the active $DVFF^2$ is capable of avoiding the obstacle, generating the tracking trajectory shown in blue. The presence of a moving obstacle in the narrow corridor seen in Fig. 4(c) also makes the controller take evasive action. The close proximity of the robot and the dynamic obstacle has as a result slow motions, the robot picking up speed once the obstacle has been cleared. This is also the case when the platform crosses narrow spaces such as doors, as forces sensed from the environment are exerted on the platform in opposed directions, effectively resulting in reduced linear speeds. For the simulations presented here the moving obstacle has been made to follow a predetermined path, tracked by a constant speed linear controller, with the objective to simulate the presence of a person moving about in the vicinity.

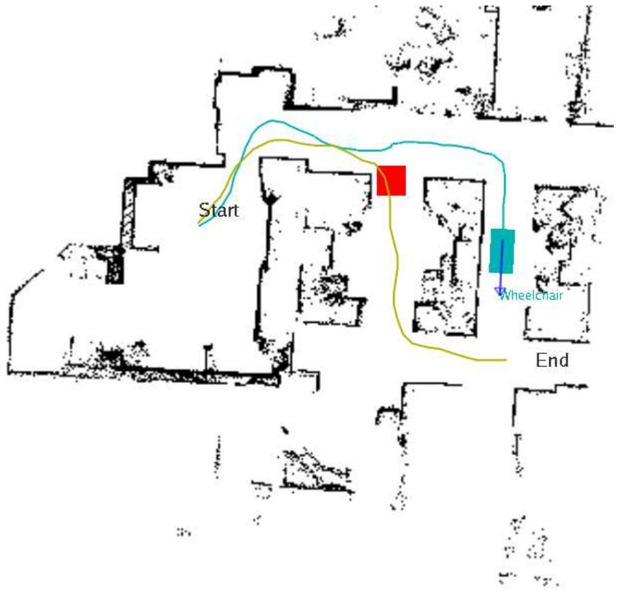
In the final scenario considered, the path the robot is made to follow is completely blocked by the presence of the static obstacle. The planner is capable of discerning this situation which triggers a complete re-planning. ICP is used to register the changes in the virtual map and the search space is modified accordingly, removing the blocked nodes and finding a new path from the robot current's location to the goal [1]. An example is displayed in Fig. 4(d).

[1]The whole re-planning process is fast, around $300ms$ in a modern PC, but the robot is nevertheless stopped for some time while it takes place for security reasons, with the view of implementing the strategy on the real platform.

(a) Initial configuration, clear of obstacles.

(b) Reacting to a new static obstacle (red) in the scene.

(c) Reacting to the presence of a dynamic obstacle (orange) in the scene.



(d) The path is fully blocked and a major re-planning is carried out.

Fig. 4. A case study of the robot navigation strategy in action in the presence of static and dynamic obstacles.

An intrinsic drawback of the proposed methodology is due to the reactive speed changes as dictated by the $DVSF^2$ controller. In can be observed how when traversing narrow passages such as doors, the repulsive force field felt by the robot produces slightly oscillating angular motions. This is because as the robot is made to closely approach the D_{min} discomfort zone, the strongest of repulsive forces are at play. The zealous safety-oriented proposed controls dictate very slow linear speeds for this scenario, which combine with the fast nature of the opposing repulsive forces and manifest in the angular controls oscillating behaviour. While this known issue is dampened on the real platform, a low-pass filter is nevertheless necessary to smooth out this undesirable behaviour.

The approach is currently being implemented on the real wheelchair platform , where more engineering work is underway to provide it with better sensing capabilities to be able to fully demonstrate the concept.

VI. CONCLUSIONS AND FUTURE WORKS

A local obstacle avoidance controller integrated within an optimised path planner has been proposed to counteract the effect of uncertainties arising from planning in an ever changing world. Based on perceptive data, and the kinematic status of robot and surroundings, a varying family of virtual force fields are generated around the moving platform in response to objects in the environment. The effect of this repulsive force (null in the absence of obstacles) is to maximise the safe motion of the robot towards the local waypoints provided by the planner, "shielding" the platform in an optimal manner and preventing collisions with the objects in the local area, static and dynamic.

An array of challenging simulated scenarios have been presented to demonstrate the good results of the proposed reactive behaviour in cluttered environments. This was made even more acute given the dimensions and mobility of the robot - an automated wheelchair - with respect to its surroundings.

Work is currently underway to implement the proposal in the real wheelchair platform. A mechanism to segmenting out the moving obstacle from its surroundings is necessary to fully demonstrate the approach, and a moving object detection and tracking strategy [38] is being developed.

ACKNOWLEDGMENT

This work is supported by the Australian Research Council (ARC) through its Centre of Excellence programme, and by the New South Wales State Government. The ARC Centre of Excellence for Autonomous Systems (CAS) is a partnership between the University of Technology Sydney, the University of Sydney and the University of New South Wales.

REFERENCES

[1] T. Taha, J. Valls Miró and Gamini Dissanayake. Sampling Based Time Efficient Path Planning Algorithm for Mobile Platforms, in *International Conference on Man-Machine Systems*, Sep 2006. (to be published)

[2] T. Taha, J. Valls Miró and D. Liu. An Efficient Path Planner for Large Mobile Platforms in Cluttered Environments, in *IEEE International Conference on Robotics, Automation and Mechatronics*, 225-230, Jun 2006.

[3] D. Wang, D. Liu and G. Dissanayake. A Variable Speed Force Field Method for Multi-Robot Collaboration, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct 2006. (to be published)

[4] J. C. Latombe. Motion Planning : A Journey of Robots, Molecules, Digital Actors, and Other Artifacts, *International Journal of Robotics Research*, vol.18, no.11, 1119-1128, 1999.

[5] J.H. Reif. Complexity of the Mover's Problem and Generalizations, in *Proc. of IEEE Symposium on Foundations of Computer Science*, New York, 421-427, 1979.

[6] J. Barraquand and J.C. Latombe. Robot Motion Planning: A Distributed Representation Approach, *The International Journal of Robotics Research 10*, no.6, 628-649, Dec. 1991.

[7] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun. The Interactive Museum Tourguide Robot, In *Proc.of the Fifteenth National Conference on Artificial Intelligence*, Madison, Wi, 1998.

[8] Chang, H. and T.-Y. Lai. Assembly Maintainability Study with Motion Planning, in *Proc. of IEEE International Conference on Robotics and Automation*, Los Alamitos, CA, 1012-1019, 1995.

[9] D. Brutlag, M.S. Apaydin, C. Guestrin, D. Hsu, C. Varma, A. Singh and J.-C. Latombe. Using Robotics to Fold Proteins and Dock Ligands, *Bioinformatics*, 18:S74-S83, 2002.

[10] T. Lozano-Pérez. Spatial Planning: a Configuration Space Approach, *IEEE Transactions on Computers*, vol. C-32, 108-120, IEEE Press, 1983.

[11] T. Lozano-Pérez and M. Wesley. An Algorithm for Planning Collision-free Paths among Polyhedral Obstacles, *Communications of the ACM*, vol.22, no.10, 560-570, Oct 1979.

[12] Y. K. Hwang and N. Ahuja. Gross Motion Planning - A Survey, *ACM Computing Surveys*, 24, no.3, 219-291, Sep 1992.

[13] D. Hsu, L. E. Kavaki, J. C. Latombe, R. Motwani and S. Sorkin. On finding narrow passages with probabilistic roadmap planners, In *Proc. of Workshop on the Algorithm Foundations of Robotics*, 141-154, A K Peters, 1998.

[14] D. Hsu, T. Jiang, J. Reif and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners, In *Proc. of IEEE International Conference on Robotics and Automation*, 4420-4426, 2003.

[15] L. E. Kavraki, P. Svestka, J. C Latombe and M. H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Trans. on Robotics and Automation*, vol.12, no.4, 566-580, 1996.

[16] V. Boor, M. H. Overmars and A. F. van Der Stappen. The Gaussian Sampling Strategy for Probabilistic Roadmap Planners, in *Proc. of IEEE International Conference on Robotics and Automation*, 1018-1023, 1999.

[17] G. Sanchez and J. C. Latombe. A Single-query Bi-directional Probabilistic Roadmap Planner with Lazy Collision Checking, in *Int. Symp. Robotics Research*, 2001.

[18] S. M. LaValle. Rapidly-exploring Random Trees: a New Tool for Path Planning, TR 98-11, *Computer Science Dept., Iowa State University*, Oct 1998.

[19] P. Cheng and S.M. La Valle . Resolution Complete Rapidly Exploring Random Trees, in *Proc. IEEE Intl. Conf. on Robotics and Automation*, 267-272, 2002.

[20] P. Choudhury and K. Lynch. Trajectory Planning for Second-order Under Actuated Mechanical Systems in Presence of Obstacles, in *Proc. of the Workshop on Algorithmic Foundation of Robotics*, 2002.

[21] J. Kim and J.P. Ostrowski. Motion Planning of Aerial Robot Using Rapidly-Exploring Random Trees with Dynamic Constraints, in *IEEE Int. Conf. Robot. & Autom.*, 2200-2205, 2003.

[22] J. M. Philips, N. Bedrossian and L. E. Kavraki. Guided Expansive Spaces Trees: a Search Strategy for Motion and Cost-constrained State Spaces, in *Proc. IEEE Int. Conf. Robot. & Autom.* , 3968-3973, 2004.

[23] S. G. Vougiouksa. Optimization of Robot Paths Computed by Randomised Planners, in *Proc. of Int. Conf. on Robot. & Autom.*, 2160-2165, 2005.

[24] J. Barraquand, L. E. Kavraki , J. C. Latombe, T. Y. Li, R. Motwani and P. A. Raghavan. Random Sampling Scheme for Path Planning, *International Journal of Robotics Research*, vol.16 no.6, 759-774, 1997.

[25] S. M. La Valle. http://msl.cs.uiuc.edu/msl

[26] S. M. LaValle and J. J. Kuffner. Randomized Kinodynamic Planning, in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 473-479, 1999.

[27] V. J. Lumelsky and T. Skewis. Incorporating Range Sensing in the Robot Navigation Function, *IEEE Transactions on Systems, Man and Cybernetics*, vol.20, 1058-1069, 1990.

[28] O. Khatib. Real-time Obstacle Avoidance for Manipulators and Mobile Robots, *Int. Journal of Robotics Research*, vol.5, 90-98, 1986.

[29] J. Borenstein and Y. Koren. Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.1, 572-577, 1990.

[30] H. Moravec and A. Elfes. High Resolution Maps from Wide Angle Sonar, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.2, 116-121, 1985.

[31] I. Ulrich and J. Borenstein. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.2, 1572-1577, 1998.

[32] I. Ulrich and J. Borenstein. VFH*: Local Obstacle Avoidance with Look-Ahead Verification, in *Proc of IEEE International Conference on Robotics and Automation*, vol.3, 2505-2511, 2000.

[33] S. Quinlan and O. Khatib. Elastic Bands: Connecting Path Planning and Control, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.2, 802-807, 1993.

[34] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.4, 3375-3382, 1996.

[35] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance, *IEEE Robotics & Automation Magazine*, vol. 4, 23-33, 1997.

[36] O. Brock and O. Khatib. High-speed Navigation Using the Global Dynamic Window Approach, in *Proc. of IEEE International Conference on Robotics and Automation*, vol.1, 341-346, 1999.

[37] J. Minguez, L. Montano and J. Santos-Victor. Abstracting Vehicle Shape and Kinematic Constraints from Obstacle Avoidance Methods, *Autonomous Robots*, 2005.

[38] C. Wang, C. Thorpe and A. Suppe. LADAR-based Detection and Tracking of Moving Objects from a Ground Vehicle at High Speeds, in *Proc. of IEEE Intelligent Vehicles Symposium*, 416-421, Jun 2003.